05

Una forma diferente de probar un if

Tenemos el siguiente código:

```
<script>
    var aprendi = true;
    if(aprendi == true) {
        alert("El instructor queda muy feliz");
    } else {
        alert("El instructor no va a desistir hasta que el alu
    }
</script>
```

COPIA EL CÓDIGO

No hay ninguna novedad ¿cierto?, inclusive ya lidiamos con códigos como ese en el curso anterior. Nuestra condición `if` prueba si el valor de la variable `aprendi` es verdadera, y para ello, usamos el operador `==` .

Sin embargo, podemos simplificar el código para:

```
<script>
    var aprendi = true;
    if(aprendi) {
        alert("El instructor queda muy feliz");
    } else {
        alert("El instructor no va a desistir hasta que el alu
```

```
    }
</script>
```

COPIA EL CÓDIGO

Observa que no usé más el operador `==`. Puede que estés intrigado por qué funciona correctamente. Pero si analizamos la sintaxis del condicionante `if()`, él espera recibir `true` o `false` para saber si ejecuta el código dentro del `if` o dentro del `else`. Si `aprendi` ya lo estamos inicializando como `true` es redundante preguntar dentro del bloque `if` nuevamente `aprendi == true`.

Las variables booleanas ya guardan `true` o `false` y podemos usarla directamente en el `if` ahorrando algunos caracteres. Sin embargo, si te sientes más seguro con la forma anterior, puedes continuar usándola sin ningún problema.